

第 5 章



嵌入式系统软件开发与调试

由 51 单片机组成的微型计算机系统常被嵌入到其他设备中辅助其应用,该系统被称为嵌入式系统。当 51 单片机电子线路设计无误后,其后续工作将是为嵌入式系统编写控制软件。本章将介绍嵌入式系统软件开发与调试的集成环境 Keil 工具,Keil 具备了汇编语言、C51、ARM C 语言源程序的编辑、编译、连接、调试、模拟执行等功能。

5.1 Keil 开发环境简介

Keil 是 Keil Software 公司出品的针对使用汇编、C51、ARM C 等语言控制、管理 Intel 51、ARM 等系列单片机的集成编辑、编译、连接、调试、模拟运行的软件开发平台。

5.1.1 Keil 开发环境主要功能

Keil 是应用于 Windows 的窗口交互式的集成开发环境(IDE),其主要功能如下。

- (1) 编辑——通过编辑窗口实现汇编(*.asm)或 C51 语言(*.c)代码的输入和修改。
- (2) 编译——C51 或 A51 编译器对源文件代码编译生成目标文件*.obj。
- (3) 连接——BL51 定位连接器将目标文件生成绝对目标文件*.abs。
- (4) 转换——OH51 转换器将绝对目标文件转换标准的机器代码格式文件*.hex。
- (5) 模拟——dScope51 调试器和模拟器可针对源级或机器代码动态调试、模拟运行。
- (6) 管理——Keil 以工程(project)项目的形式管理所开发的应用程序。

5.1.2 Keil 开发应用程序流程

使用 Keil 集成开发环境开发 51 单片机应用程序流程如下。

- (1) 创建一个工程项目文件,有关应用程序的所有文档都包含在该工程中。
- (2) 选择 51 单片机芯片型号,应用程序是针对该型号的单片机开发的。
- (3) 设置工程选项,使其符合应用程序开发要求。
- (4) 根据应用程序需求,选择添加单片机启动代码、管理系统等到工程项目中。
- (5) 创建源程序文件,输入编辑源程序文件,并添加到当前工程项目组中。

- (6) 编译源程序文件,修改编译错误。
- (7) 单步或连续模拟运行调试应用程序,通过调试窗口观察应用程序运行结果。
- (8) 应用程序运行结果正确,Keil 生成的 *.hex 即是写入单片机程序存储区的文档。

5.1.3 Keil 开发环境界面

Keil 应用于 Windows 操作系统的软件 Keil μ Vision 启动后其窗口界面如图 5-1 所示,Keil 集成环境主菜单项有 File(文件)、View(视图)、Project(项目)、Flash(下载)、Debug(调试)、Peripherals(外围设备)、Tools(工具)、SVCS(版本控制)、Window(窗口管理)、Help(帮助)等,Keil 的主要功能都可以从主菜单中引用,但为更加方便、快捷,主菜单下设有快捷操作按钮工具栏,工程项目中有右键菜单。

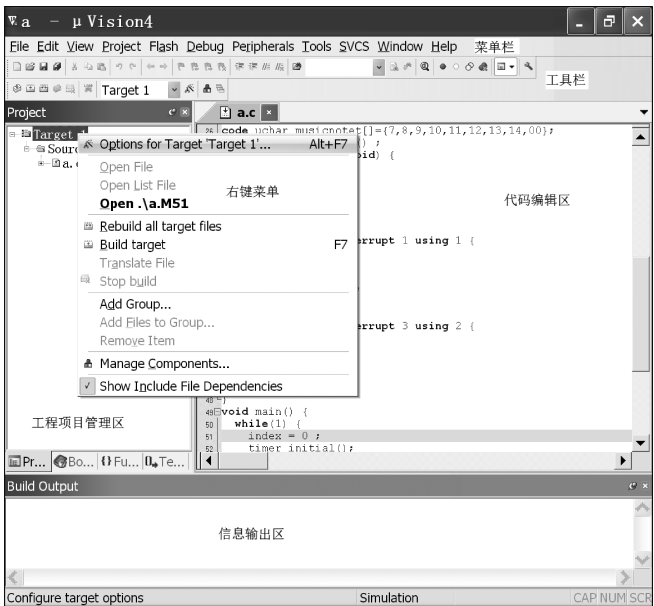


图 5-1 Keil 集成环境窗口界面图

5.2 在 Keil 环境中开发应用程序

在 Keil 中开发应用程序是以工程项目为单元的,首先需要创建一个针对某款单片机型号的工程项目,其次在工程项目中添加应用程序代码,最后编译输出可执行代码。



视频讲解

5.2.1 在 Keil 环境中创建工程项目

在 Keil 环境中可以创建多个工程项目(Target),每个项目可以创建(包含)多个源文件

组(Source Group),每个源文件组中可以添加多个应用程序源代码文件,创建一个工程项目(包含一个源文件组)并在一个源文件组中添加应用程序源代码的操作步骤如下。

- (1) 选择 Keil 主菜单 Project 中的 New μ Vision Project 菜单项。
- (2) 在弹出的如图 5-2 所示的 Create New Project 对话框中为工程项目命名,并保存为工程项目文件。



图 5-2 创建新的工程项目界面图

- (3) 在后续弹出的“Select Device for Target...”对话框中选择一款单片机型号(或兼容型号),如图 5-3 所示,对所选单片机的描述见右边窗口(Description)。

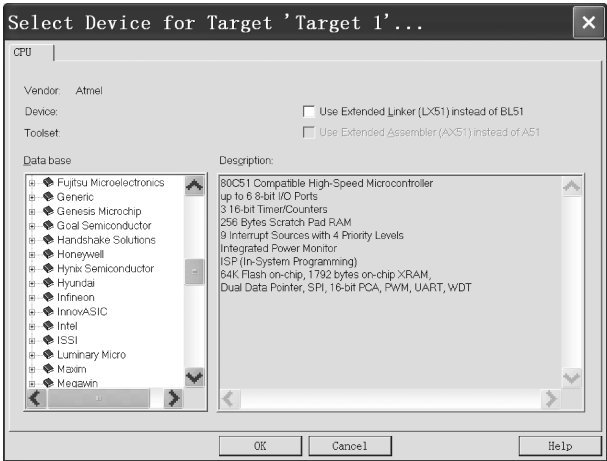


图 5-3 选择单片机型号对话框

- (4) 单击“Select Device for Target...”对话框中的 OK 按钮,Keil μ Vision 将会自动为所选择单片机添加合适的、标准的启动代码,弹出如图 5-4 所示的对话框——提示是否将该机型的启动代码添加到工程项目中。

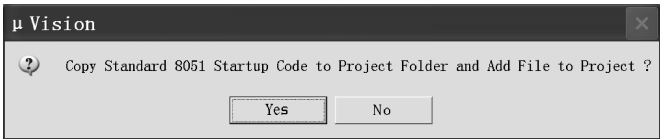


图 5-4 选择添加单片机启动代码

(5) 在工程项目中需要使用单片机的启动代码时,则单击 Yes 按钮,但要对该启动代码的初始化工作有所了解,通常该段代码以汇编语言形式(例如,针对 51 单片机文件名为 STARTUP.A51)被添加到工程项目中。

(6) 将应用程序源代码文件添加到工程项目源程序组(Source Group)中,在 Project 中 Source Group 1 的右键快捷菜单中选择“Add Files to Group...”菜单项,如图 5-5 所示。

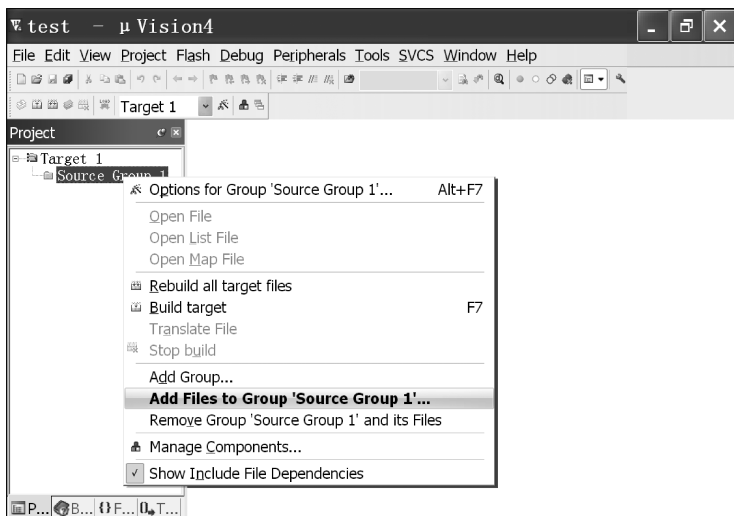


图 5-5 添加应用程序源代码文件

(7) 在如图 5-6 所示的“Add Files to Group...”对话框中,选择汇编语言(*.s*、*.a*)或 C51 语言(*.c)的应用程序源代码文件添加到工程项目中。

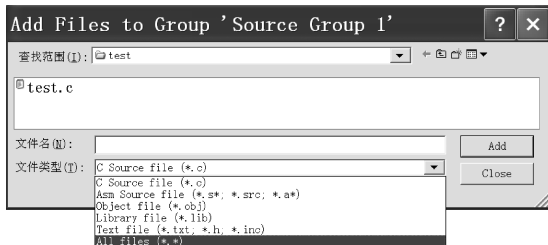


图 5-6 选择汇编或 C51 源文件

(8) 如果 C51 语言程序中有自定义(非 Keil 集成环境提供)的头文件(*.h),则需要将之添加到工程项目中,当应用程序所需所有文件都添加到工程项目后,针对某一款单片机的工程项目创建完毕。

5.2.2 在 Keil 环境中编译工程项目

Keil 环境集成了多款编译器,面向、支持 Intel 51、ARM 系列单片机的汇编语言和 C51、

ARM C 语言程序源代码实施编译操作,将程序源代码转换为单片机的可执行代码(BIN 二进制代码)。Keil 集成环境的编译操作分为两步:一是将程序源代码编译为目标代码(Obj);二是由连接器将目标代码连接为可执行代码,一般情况输出要求为 HEX 格式文件(*.hex),该格式文件是 Intel 公司制定的,它是由十六进制数组成的机器码或数据常量文件,多数单片机系统仿真器使用、支持 HEX 文件格式实施仿真操作,HEX 格式文件可由转换器(HEX→BIN)转换为 BIN 二进制格式文件(可执行代码)后写入单片机程序存储区。

Keil 环境的编译操作通常是针对工程项目而言的,在实施编译操作之前,应根据单片机型号等参数对工程项目做必要的设置,其设置与编译操作步骤如下。

(1) 选择 Keil 主菜单 Project 中 Options for Target 菜单项,弹出“Options for Target…”对话框,如图 5-7 所示。

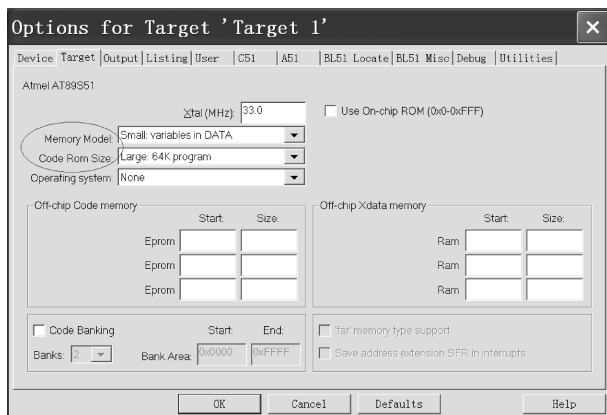


图 5-7 Target 设置对话框

(2) 在“Options for Target…”对话框中选择 Target 选项卡,根据实际开发的单片机系统设置 Target 选项卡中的各项内容,例如,“Memory Model”存储器模式设置中有 3 个选择: Small 模式——变量存放单片机内部 RAM 区(128B)、Compact 模式——变量存放单片机外部 RAM 区(256B)、large 模式——变量存放单片机外部 RAM 区(64KB),依据硬件情况选择一个存储器模式。

(3) 在“Options for Target…”对话框中选择 Output 选项卡,如图 5-8 所示,设置编译连接后输出 HEX 格式文件,选中“Create HEX File”复选框,另外,在该选项卡中通过单击 Select Folder for Objects 按钮可以设置输出文档的存放目录。

(4) 如果在编译和连接时有特殊要求,或使用其他编译器和连接器,可通过“Options for Target…”对话框中的 C51、A51、BL51 等选项卡完成设置。

(5) 编译连接设置符合单片机系统要求后,选择 Keil 主菜单的 Project→Build target 菜单项(工具栏中有编译快捷按钮或 F7 键),也可以选择 Rebuild all target files 菜单项完成对整个工程项目的编译和连接。当编译连接出现错误时,错误信息输出到信息框中,提示

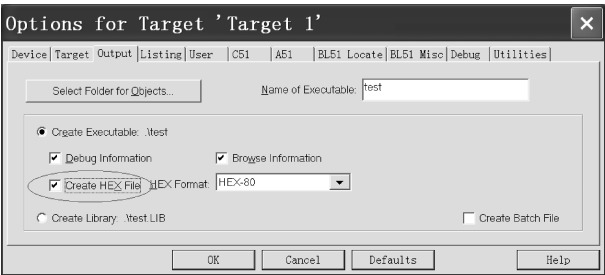


图 5-8 Output 选项卡

错误类型和行号,常见错误信息如表 5-1 所示;当编译连接无误时,在信息框中输出的信息如图 5-9 所示,完成对工程项目的编译和连接。

表 5-1 Keil 编译器常见编译错误列表

错误信息	错误说明
Argument list syntax error	参数表语法错误
Bad character in paramenters	参数中有不适当的字符
Bit field too large	位区域太长
Call of non-function	调用未定义的函数
Code has no effect	代码不可能被执行
Compound statement missing“{”	分程序漏掉“{”
Declaration syntax error	声明语法错误
Division by zero	除数为零
Expression syntax error	表达式语法错误
Function definition out of place	函数定义位置错误
Function should return a value	函数需要有返回值
Hexadecimal or octal constant too large	十六进制或八进制常数太大
Illegal initialization	非法的初始化
Illegal use of floating point	非法的浮点运算
Improper use of a typedef symbol	类型定义符号使用不恰当
In-line assembly not allowed	不允许使用行汇编
Incompatible storage class	存储类别不相容
Incompatible type conversion	不相容的类型转换
Incorrect number format	错误的数据格式
Invalid indirection	无效的间接操作
No declaration for function “xxx”	没有函数 xxx 声明
No stack	缺少堆栈
Not a valid expression format type	不合法的表达式格式
Not an allowed type	不允许使用的类型
Out of memory	内存不够用
Possibly incorrect assignment	赋值可能不正确

续表

错误信息	错误说明
Redeclaration of “xxx”	重复定义 xxx
Register allocation failure	寄存器寻址失败
Statement missing “;”	语句后缺少“;”
Too few parameters in call	函数调用时实参数少于形参数
Too many error or warning messages	错误或警告信息太多
Type mismatch in parameter “xxx”	参数 xxx 类型不匹配
Undefined symbol “xxx”	没有定义的符号 xxx
Void functions may not return a value	Void 类型的函数没有返回值



图 5-9 编译链接成功后的输出信息

5.3 在 Keil 环境中调试运行

一个工程项目在 Keil 环境中编译连接通过后,说明工程项目中源代码程序没有语句或语法等错误,但程序是否可实现预期结果则需要被执行后才能判断。Keil 环境提供了模拟连续或单步运行程序的功能,Keil 环境的各种调试窗口可观察到程序执行的结果,通过模拟运行程序可查找出程序运行时出现的错误,同时可返回程序源代码中加以修正。

5.3.1 Keil 环境调试前的设置

在 Keil 环境中模拟运行程序前需要完成必要的设置,首先根据单片机系统的实际工作频率设置模拟运行时使用的主频,单击 Keil 主菜单 Project→Options for Target 菜单项,在弹出的“Options for Target…”对话框中选择 Target 选项卡,在 Xtal(MHz)输入框中输入模拟运行时使用的主频,如图 5-7 所示;其次在“Options for Target…”对话框中选择 Debug 选项卡,如图 5-10 所示,依据实际单片机系统以及源程序情况(是否添加启动代码等)设置各个选项。

5.3.2 Keil 环境调试主界面

在 Keil 主菜单中选择 Debug→Start/Stop Debug Session 菜单项(快捷键 Ctrl+F5)进入程序模拟运行调试主界面,如图 5-11 所示。

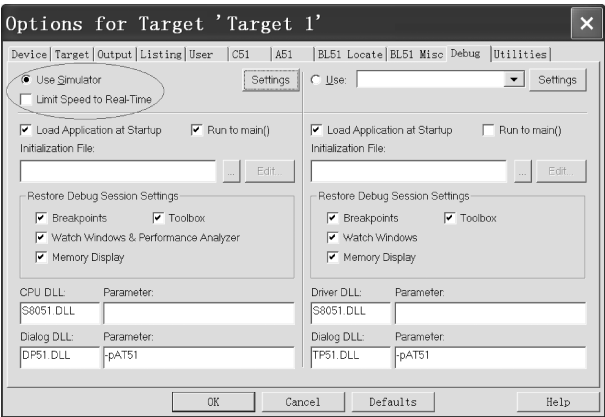


图 5-10 Debug 选项卡



图 5-11 Keil 环境调试主界面

在调试主界面中有调试操作按钮工具栏,方便调试操作;有源程序和反汇编程序代码窗口,方便观察程序将要执行的语句;有 CPU 寄存器窗口,方便查看寄存器内容;有源程序符号窗口,方便定位查找;有程序调用窗口,方便观察调用情况等。

5.3.3 Keil 环境调试操作

在 Keil 环境的模拟运行调试主界面中,其调试操作命令在 Keil 的主菜单 Debug 中,也可通过工具栏按钮或快捷键控制程序的运行以及在程序的运行途中设置断点等操作,Debug 菜单、工具栏主要按钮、快捷键的操作功能如表 5-2 所示。

表 5-2 Keil 模拟运行调试环境主要操作键表

Debug 菜单项	快捷 键	按钮 图标	功 能 说 明
Reset CPU			CPU 执行程序复位
Run	F5		连续运行程序
Stop			停止运行程序
Step	F11		单步(每条指令)跟踪执行程序
Step Over	F10		单步(函数或子程序为 1 步)执行程序
Step Out	Ctrl+F11		退出单步跟踪执行程序
Run to Cursor Line	Ctrl+F10		程序执行到光标处
Show Next Statement			显示将要执行的下一条语句
Breakpoints...	Ctrl+B		显示设置断点对话框
Insert/Remove Breakpoint	F9		插入/消除断点
Enable/Disable Breakpoint	Ctrl+F9		断点有效/失效
Disable All Breakpoints			所有断点失效
Kill All Breakpoints	Ctrl+Shift+F9		解除所有断点

5.3.4 Keil 环境调试窗口

在 Keil 环境的调试界面中已经默认安排显示了一些调试窗口,通过在 Keil 主菜单中选择 Window→Reset View to Defaults 菜单项,可设置为默认显示位置,如图 5-11 所示,但这些调试窗口支持各种显示方式,任意安排窗口所显示的位置,方便观察。Keil 调试环境的所有调试窗口都可以通过选择 Keil 主菜单 View 中的菜单项打开,也可以通过工具栏按钮快速打开常用的、重要的调试窗口,调试窗口工具栏如图 5-12 所示。



图 5-12 Keil 环境打开调试窗口工具栏

1. 符号浏览窗口 Symbol

在 Keil 调试环境主菜单中选择 View→Symbol Window 菜单项,可打开符号浏览窗口。该窗口显示工程项目中的各种符号名称,包括专有符号,用户自定义符号(函数名、变量名、标号)等,在 Name 显示的符号名处右击则弹出右键快捷菜单,通过相应命令可将该符号导出到其他需要观察的窗口中,如图 5-13 所示。

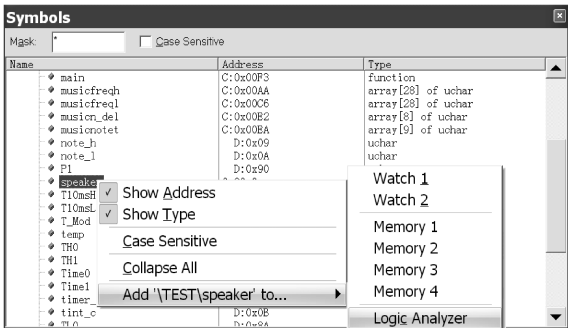


图 5-13 符号浏览显示窗口

2. CPU 寄存器窗口 Registers

在 Keil 调试环境主菜单中选择 View→Registers Window 菜单项,可打开 CPU 寄存器窗口,该窗口将动态显示程序运行时 CPU 寄存器某个时刻的内容,如图 5-14 所示。

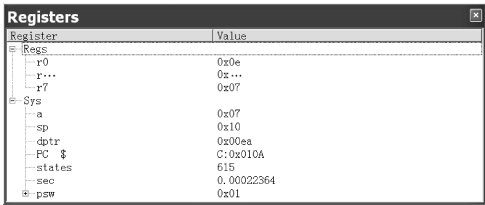


图 5-14 CPU 寄存器显示窗口

3. 观察窗口 Watch

在 Keil 调试环境中选择 View→Watch Windows→Watch1 菜单项,可打开观察窗口,在观察窗口中双击 Name 选项或按 F2 键可添加需要查看的变量等,在 Value 显示栏中动态显示程序运行时该变量某个时刻的内容,如图 5-15 所示。另外,右击 Name 栏中的内容弹出右键快捷菜单,通过 Number Base 项可设置数据显示格式,通过“Add...to...”项可将 Name 栏中的符号转送到其他显示窗口。

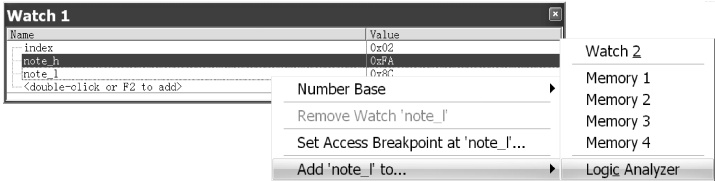


图 5-15 观察显示窗口

4. 存储器窗口 Memory

在 Keil 调试环境中选择,View→Memory Windows→Memory1 菜单项,可打开存储器窗口。Keil 调试环境提供 4 个存储器窗口,在存储器窗口的“Address:”输入框中输入“存

储区代码：首地址”，则存储器窗口将从“首地址”开始逐一显示存储器中的内容，“存储区代码”有 D(单片机片内 RAM 直接寻址区 data)、I(单片机片内 RAM 间接寻址区 idata)、X(单片机片外 RAM 数据区 xdata)、C(单片机程序存储区 ROM code)等，右击存储器窗口区数据则可以选择显示的数据格式，如图 5-16 所示。如果修改存储器内容，则双击数据区的数据(或选择右键菜单中的 Modify 项)实现数据修改。

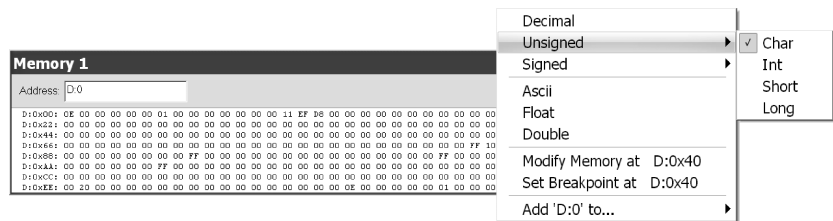


图 5-16 存储器显示窗口

5. 串口模拟终端窗口 Serial

在 Keil 调试环境中选择 View→Serial Windows 菜单项，可打开串口模拟终端窗口，显示通过串口传输的数据。另外，多数单片机的数据传输是通过串口实现的，C51 语言函数 printf() 的输出内容则显示到该窗口中，通过其右键菜单可选择数据传输格式等，如图 5-17 所示。

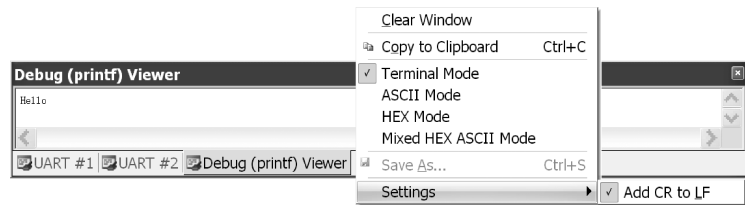


图 5-17 串口输出显示窗口

6. 逻辑分析仪显示窗口 Analysis

在 Keil 调试环境中选择 View→Analysis Windows→Logic Analyzer 菜单项，可打开逻辑分析仪窗口，在该窗口中通过单击 Setup 按钮输入需要分析和观察的量(符号名称)，或从其他窗口导入各种量，逻辑分析仪将记录程序运行期间所有时刻各个量的状态以及时序。在程序停止运行时，可以通过逻辑分析仪窗口查看各个量在运行期间的情况，如图 5-18 所示。

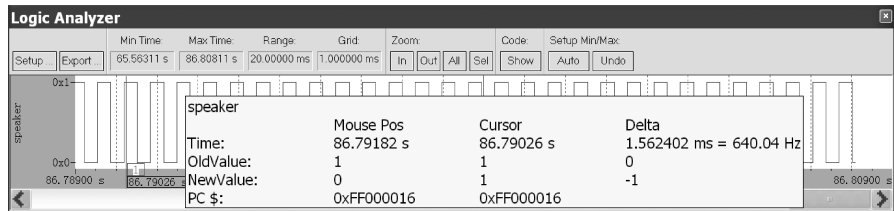


图 5-18 逻辑分析仪显示窗口

5.3.5 Keil 调试环境中设置断点

在应用程序调试过程中,有时候不需要单步运行,观察每一条指令的运行结果,而是希望程序运行到某条指令后停止运行,观察程序运行到该指令处的运行结果,因此,需要在停止运行指令处设置断点,将光标移动到设置断点处,在 Keil 调试环境中选择 Debug→Insert/Remove Breakpoint 菜单项(或快捷键 F9),实现插入断点,如图 5-19 所示。

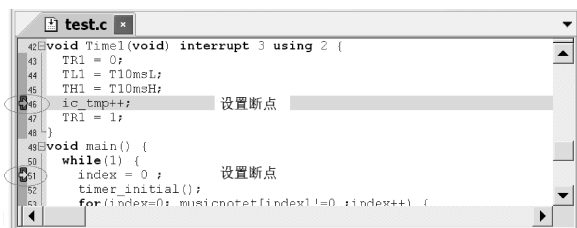


图 5-19 在源程序中设置断点

在 Keil 调试环境中选择 Debug→Breakpoints 菜单项(或快捷键 Ctrl+B),在弹出的断点(Breakpoints)对话框中可实现对断点的设置、删除等管理操作,如图 5-20 所示。

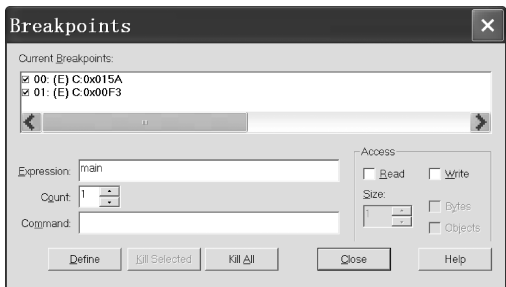


图 5-20 断点管理对话框

5.3.6 Keil 调试环境中可编程接口设备

在单片机系统中 CPU 外围一般配有多种可编程接口设备(外围设备),使用汇编语言或 C51 语言编写的应用程序除指挥、控制 CPU 工作外,还可以通过 CPU 对外围设备实施控制或操作,标准的外围设备有输入/输出(I/O)、中断管理、定时器/计数器、串口等接口,在 Keil 环境处于调试状态时,这些外围设备都可以以独立窗口的形式打开。在 Keil 调试环境中选择 Peripherals 菜单中的 I/O-Ports、Interrupt、Timer、Serial 等菜单项打开对应的窗口,在应用程序调试运行过程中,通过窗口操作实现对可编程接口设备的控制。

1. 输入/输出并口控制窗口 Parallel Port

单片机系统一般有多个输入/输出并口,对于普通 51 单片机系统有 4 个并口(P0~P3),在 Keil 调试环境中选择 Peripherals→I/O-Ports 菜单项,选择打开某个并口,如

图 5-21 所示(打开 P0 并口),通过该窗口可以控制某位为 1(√)或 0。

2. 中断控制窗口 Interrupt System

在 Keil 调试环境中选择 Peripherals→Interrupt 菜单项,打开中断管理、控制接口,如图 5-22 所示。当选择 Int0 中断源操作 EX0(√)项时,可实现该中断源的开中断操作。

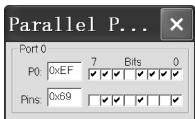


图 5-21 输入/输出并口控制窗口

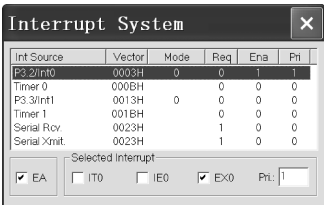


图 5-22 中断控制窗口

3. 定时器/计数器控制窗口 Timer/Counter

单片机系统一般有多多个定时器/计数器,对于普通 51 单片机系统有 2 个定时器/计数器(T0、T1),在 Keil 调试环境中选择 Peripherals→Timer 菜单项,选择打开某个定时器/计数器接口设备,如图 5-23 所示(打开 T0),在该窗口中可以选择定时器/计数器的工作模式、设置定时器/计数器的初始值以及启动 TR0(√)或停止定时器/计数器等操作。

4. 串口控制窗口 Serial Channel

在 Keil 调试环境中选择 Peripherals→Serial 菜单项则打开串口管理、控制接口,如图 5-24 所示。在该窗口中可以选择串口工作模式、设置串口发送 TI(√)或接收 RI(√)中断等操作,以及观察串口传输的数据(SBUF 内容)等。

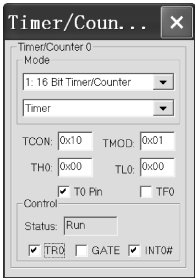


图 5-23 定时器/计数器控制窗口



图 5-24 串口控制窗口